ARGONAUTS

Vol 2, 1

for the Amiga Developer/Entrepeneur

Morning Star International

Table of Contents

HotLinks for Today and Tomorrow

Dan Weiss Soft-Logik

"No man is an island..."

There was a time when having a computer meant having an electricity gobbling behemoth that had its own room, air conditioning and possibly plumbing. It was attended by servants that fed it punched cards and magnetic tape. As time passed the monsters grew smaller but still demanded special treatment. In return for the special care and steady diet of high voltage, the monsters allowed mere mortals to speak to them via remote terminals. Even so, each program running performed a separate task, and getting them to work together involved complex batch programs.

In the early '80's a dramatic change took place in computers. The giant computers were broken down into many smaller computers. Each smaller computer had less power, but was meant for individual, or personal use. Hence the advent of personal computers. Since the inception of the personal computer, these small machines have been striving for the power of the old monsters. As the power of each individual machine grows, more of the monster computer features are added. In the case of the Amiga, multi-tasking was claimed from the monsters. In many other computers, networks were used to simulate the terminals/machine model of the monsters.

Now finally, personal computers are reaching beyond their predecessors. No longer are programs limited to working by themselves, now they can work with each other. Up till now only simple fledgling steps have been taken; Publish and Subscribe on the Apple Macintosh, DDE and OLE in Microsoft Windows. Finally the first real steps have been taken with HotLinks on the Amiga.

What is HotLinks

Plain and simple, HotLinks is a system to exchange data between multiple programs while they are running, and even when they are not. HotLinks is not a simple "pipe" from one product to another, but rather a system that provides a common user interface and a well defined method of interaction. Plain and simple HotLinks lets programs work together.

To meet a need

Many people ask "why do I need HotLinks". Perhaps the broader question of "why does anyone need HotLinks," would be better. HotLinks was developed to meet a need that many programs have. Gone are the days when a major program only deals with one kind of data, and all data is created in the program. Today word processors, spreadsheets,

Argonauts is published quarterly by

Morning Star International
24 - 50 26 street
Astoria NY 11102

Publisher Marc P Seybold

Address all correspondence to:

Argonauts
Box 94
Pearl River NY 10965-0094

Postmaster: Send address changes to Argonauts, Box 94, Pearl River NY 10965-0094

Many of the designations used by manufactures and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this newsletter and we were aware of a trademark claim, designations have been printed in initial caps.

Argonauts is not affiliated with Commodore Business Machines.

Subscription Information.

$49.95 United States
$56.00 All other countries
Subscriptions are 4 issues a year
Non US subscriptions are air mail delivered

Make checks or money orders payable to:
Morning Star International.
U.S. funds drawn on a U.S. bank ONLY.

Mail to: Argonauts att: subscriptions
Box 94
Pearl River NY 10965-0094
USA

databases, presentation packages and page layout programs all combine parts and pieces from other programs. The problem is that when you make changes to the data in one program, you need the data in the other programs to be properly updated. Of course this can be done by hand, but it tends to be a long and tedious project. Logically this is something that the computer should be able to "just do". Well it can now, with HotLinks.

HotLinks offers a system that keeps track of changes made to files and notifies the programs that use them of the changes. Most importantly it works with any Amiga program and it is all transparent to the user. Many programs try to simulate this using ARexx scripts. While ARexx would seem to offer more flexibility, it doesn't since no two developers seem to be able to agree on how to implement the features HotLinks has via ARexx. Even worse the user is often faced with the daunting challenge of bending ARexx to their will. While ARexx is a wonderful tool for the control of programs, it is not ideally suited for the task of exchanging and maintaining information.

Beyond Linking

HotLinks goes beyond the simple idea of allowing files to be transferred between programs in three ways. First, HotLinks notifies all programs that ask to be notified of updates to the file, not just a single program that is "linked". Secondly, HotLinks maintains a database of information about the file such as, who created the file and when it was updated. Finally, the links established by HotLinks do not go away when a machine is rebooted. The information about each file is maintained on the disk so that a program that wasn't running when the change was made can still know a change was made even days, weeks or years later.

HotLinks also goes beyond simple ARexx links in that it offers a consistent interface to the user no matter what program it is accessed from. This not only reassures the user that they are using the same feature in all programs, but also removes the need to re-learn linking in each program.

Peer to Peer

Linking technologies available on other computer platforms take a limited view of what users will want to do with their data. In the case of the Apple Macintosh, the publisher of data is the only one allowed to modify the data. Any holders of the data are merely slaves to the master data. In the case of HotLinks, all programs are viewed as peers. While one program may originate a piece of data, it may not be the only program that can update that data. Instead of forcing the user to reload and modify the data in the original program, HotLinks turns control over to other programs as needed. The one worry with this system is that two programs may try to modify the data at once. HotLinks keeps this situation from happening by the careful use of modification permission controls called write locks. By allowing programs to work as a team, the Amiga begins to explore its true potential as a multi-tasking machine.

Editorial

By request, Argonauts is extending free advertising to small box adds as well as continuing the policy of free classified. See back cover for details. Note that both these offers are open only to subscribers.

Argonauts aims to serve and provoke. If the community is talking about the topics raised in an issue than I know we're on the mark. It's about cooperation. In many cases, collective action by many small companies can be as effective as unilateral action by one large company. It's about utility.

Unlike consumer oriented magazines this newsletter is about trying to make business better for all of us. One topic that we will try and address is intelligence, not IQ but information in the sense that the military uses the term. Working with as many people as want to get involved we want to publish quarterly figures showing estimated unit sales of various product categories in the Amiga market.

If the mail order houses want to provide figures we will also publish consolidated "best" seller lists. Right now the average Amiga vendor is flying blind on this kind of data. It is very difficult to make effective business decisions without having reasonably accurate figures on the size of various market segments.

Needless to say it will be impossible to publish this kind of data if most vendors do not volunteer it. Only consolidated figures would be published For example, you would know that 5,000 desktop publishing packages were sold last quarter and that they ranked PageStrea, ProPage and Saxon. We would not however publish the unit sales figures for individual products.

Amiga companies need to start exploiting their position as outsiders.This is particularly true in the program development arena. The Mac and PC world have enormous resources tied up in C++ and are in no position to make radical changes.

Amiga developers are in a position to bypass these early OOP languages and make a quantum leap. We can look for that language and environment that will make it possible for one or two programmers to turn out a complete application in six months. See future articles on the Dylan language for a con-

Part of the system

In the case of Macintosh's Publish and Subscribe, and Windows DDE and OLE, linking is a system function. So why then isn't HotLinks? Over the history of the Amiga, many third party products have enhanced the OS. This is in large part because the way the Amiga has grown has been dictated by its users and developers. As a demand for better cross platform integration brought CrossDos, so in effect has the need for better cross program integration brought HotLinks. HotLinks can serve a programmer or a user like a system function because it is open and documented like a system function. Unlike other linking systems, HotLinks is not proprietary in any way. The future development of HotLinks is open to, and in fact relies on the interaction of Amiga developers and their applications of the technology.

Beyond the machine

The next and most logical step for HotLinks is to reach from programs on one computer to those on another. Due to the simple and consistent interface, little will change for the user, but now they will be able to reach out to other machines on local the network. And while the user is reaping the benefits of consistency, developers will as well since network HotLinks and single user HotLinks use the same API. Unfortunately network HotLinks is not available yet, but is the obvious extension of current HotLinks technology.

Concept to Code

So where does all this place a developer looking for a competitive edge? HotLinks offers a way to offer greatly increased functionality with little effort. Due to the common interface and well defined data formats HotLinks offers few implementation problems to developers. Unlike putting in yet another import or export filter to support another file format, HotLinks opens the gateway to many programs with a single simple set of modifications.

Actual implementation of HotLinks is eased by the HotLinks developers kit. The kit is already available free of charge on the major on-line services in electronic form. A printed version with floppy disk is available from Soft-Logik as well. The developers kit includes HotLinks documentation, example code for implementing HotLinks and all needed pragmas and stubs.

HotLinks for today and tomorrow

The bottom line on HotLinks is that it is offering a competitive advantage for Amiga users and developers today. With future developments in the Amiga and HotLinks, live data exchange will become another hallmark of the Amiga.

The time to jump in is now.✔

Soft-Logik

1131 South Towne Square, Suite F

St. Louis MO 63123

(314) 894 - 8608

tender

Take a look at the book section in this issue for some ideas on how to make incremental changes. The ideas from these books can be phased into projects that are in progress. If you have your manual all written you won't rewrite it at the last minute. You could however change the layout. If your software program is half finished you can still make cosmetic changes. If you are starting out on a new project you can exploit all of the techniques discussed in these books.

The point is that if you start now, 1 to 2 years down the road, your products will be better across the board than if you had done nothing. Phased in incrementally, they should vastly improve quality without having much impact on development costs.

The article on package art is another good example. Getting a consensus to do something like this in the Mac and PC market would be impossible. Too'many companies and too many egos. The smallness of the Amiga market for once is an advantage. It makes something like this doable.

Finally, if you've been receiving Argonauts for free this is your last issue. Those wanting to continue the voyage will find subscription information below the masthead. ❑

Brief Notes from the Winding Road

Eric G. Young

ASF Software Corporation

During the my first year as a physics graduate student at Michigan State, I worked on software simulation of the DO detector at Fermi National Accelerator Laboratory. The principle tool for this effort was a massive body of FORTRAN called GEANT developed by French physicists. Geant is French for giant and that's exactly what this was.

The other graduate student working with me on the simulation had a multi-inch listing with yellow stickies sticking out to identify the subroutines that concerned us. It was my responsibility to write the routines that described detector components.

Not all of the frustration resulted from GEANT. The detector uses both copper and denatured uranium plates. Per American practice, the copper plates were rolled to a thickness measured inches. But, the uranium plates were rolled to a thickness measured in centimeters. This made telling GEANT how to assemble the pieces rather time consuming. However, the bulk of effort and frustration was consumed by the write-compile-link-run cycle applied to a giant body of software.

The above is a round about way of introducing a conclusion that I had reached before working with GEANT: It ought to be possible to design a computer system that will allow real time development of code without the frustrations of tracking down compiler and linker errors.

Starting Out

After graduating with my masters, I founded a company, then called Eta Software Corporation and now called ASF software corporation to develop such a system. Thus began a five year odyssey among the fringes of computer technology.

The original objective was to develop an operating system called the Advanced Productivity Programming Environment (APPE). Since I was concerned about being confused with a simian, I changed the name to Advanced Symbolic Form or ASF.

I started to write the system for IBM PC machines and to generate x86 code in real time. After several months exploring how to do this, I decided that the x86 architecture is enough of a hash that this was hopeless. I decided to design a microprocessor architecture specifically for ASF and implement this architecture as a software emulation using x86 code.

Once the emulator was written I could implement my real time code generator. From the very beginning of the effort (1987), I had decided to implement ASF using the object oriented paradigm. Code is encapsulated in objects of a class called executable records (exr for short). Variable names are provide by instances of the variable name set class.

The processor, the x9000, encapsulates all object handling in its instruction set. All the programmer needs to know about an object is its selector, a virtual memory pointer. Given the selector, the x9000

Construction Kit Follow-up

Last issue's CK article has already produced results. Eric Young's article on ASF provides a potential solution to persistent object storage on the Amiga. Sarah Bell of StepStone Corp (the company that commercialized Objective-C) will have an article in the Spring issue on porting Objective-C as a business opportunity.

Last but by no means least is Apple's Dylan language. Dylan was publicized after the first issue went to press. It is already where the Xlisp project described last issue would have ended. Dylan was designed for programmers by programmers. These people have already gone through Object-Pascal, C++, Smalltalk and Common Lisp/CLOS. Dylan is what they came up with as the next step to do the kinds of things discussed in the CK article.

What does this have to do with Amigadom? Quoting from the Dylan reference manual, "We would like to see the language spread beyond the Apple community." The Dylan reference manual is available free by writing to the address at the end of this article. We are working with Andrew Shalit at Apple Cambridge to get a Dylan article in a future issue. In the meantime you can get the reference manual now.

The down side of Dylan is that Apple themselves didn't have the in house resources to implement it. The manual preface describes how they acquired a small lisp company who in turn did the actual implementation. The open question is whether the Amiga community has the technical talent to implement a commercial quality Dylan compiler? Perhaps this is a job for the Europeans?

The arrival of 32 bit processors, AGA chips and coming DSP chips are in stark contrast to the development tools available. Long term survival requires a quantum leap beyond C. Dylan when combined with a persistent object file system and extensive class libraries would give us this quantum leap.

Objective-C could be used as a transition to get from where we are today to where Dylan could take us tomorrow. This would make the leap less disruptive.

Dylan also has an interesting ulterior attraction from an Amiga developers point of view. Eventually the installed base of AGA hard disk equipped machines will attract the atten-

locates the object in RAM or on disk, brings it into RAM if needed, and returns the real memory address of the object. The x9000 is the kernel of an object oriented operating system.

The real time generation of code is accomplished by having objects which specify the implementation of an instruction. The concept is straight forward. The instruction is conceived as a set of fields. Each field has a limited number of valid values or a range of valid values.

For those fields with a small set of possible values, the programmer selects the desired option from a menu. No more looking through a manual to see what the various values represent. For those fields which have a range of possible values, the programmer enters a numerical value. In early implementations this is a hexadecimal value.

The Prototype

The first prototype for ASF was called exr.ed for executable record editor. Exr-ed was limited by design. It allowed the development of code objects but not data objects. The significance of exr-ed is that it uncovered conceptual limitations that would have not been found otherwise.

Exr.ed was sufficient to support the development of a more sophisticated prototype called ook or object oriented kernel. Ook was designed to handle both data and code objects. It handles most data objects very nicely. Ook still has limitations. Ook does not adequately handle those situations were a member repeats a variable number of times from instance to instance, and the repeats are not all the same size. Ook treats development of code - which is the outstanding example of the doubly variable member phenomenon - as a special case.

In spite of its limitations, ook is nicer to use than Borland's IDE for MS-DOS. Those who have used IDE know that it's no toy. Ook will serve nicely as a beta version and a demonstration prototype for eliciting feedback.

An OOP CPU

Ironically the x9000 appears to have more marketing appeal than the operating system it was designed to support. As the kernel of an operating system, it can run more than one operating system concurrently. This is implemented by adding tasks to the task list object. The only possible hitch is the obvious one, I/0 conflicts.

I have begun talking to companies with established operating systems about porting their operating systems to the x9000. So far no resistance to this path has been encountered, which suggests something nice is going on.

The x9000 will be produced as cell block ASIC. The x9000 instructions are self decoding. The time to identify an instruction is nearly flat. The first derivative of the time-to-decode versus instruction-count decreases exponentially. This renders the advantage of RISC over CISC obsolete.

When I first started figuring out how to implement x9000 as a chip, I didn't understand what microcode was. Thus, the x9000 will rely exclusively on microparallelism with no microcode. This means that tion of hard pressed second tier windows and Mac developers. When, not if they start making incursions into the Amiga market, things will get interesting. They will be lean, mean, battle hardened organizations.

If they found Dylan the standard development environment, it would be in their interest to buy, merge or contract out product ports to existing Amiga companies. In contrast, using C++ as the Amiga's next generation development language will not result in any quantum reductions in development time. It will however guarantee that when the incursions begin, these companies won't have much need of existing Amiga companies.

People worry about Microsoft, Lotus, Borland and Wordperfect. How happy would you be if you had to compete against; Write-Now, ReadySetGo!, IntelliDraw or Canvas? Are the 3D guys so sure they could hold their own against say PIXAR? Something to think about. ❑

Apple Computer Inc.
Att: Dylan Manual
1 Main Street, 7th Floor
Cambridge, MA 02142

the x9000 is not a RISC or CISC processor emulated by an externally invisible sub- RISC architecture. It remains to be seen, but I strongly suspect that this is a positive factor.

The x9000 has a significant long term implication. Since the x9000 is an operating system kernel, why keep the operating system? Indeed many end users have no direct motive to buy an operating system. They buy the operating system because it supports the application they need.

This poses a tricky question: what replaces the operating system! My answer is that the present trend toward software tools and object libraries does, and this evolves in a manner that can not be foreseen. The notion of construction kits presented in the first issue of Argonauts fits in very nicely here. I have found it useful to compare objects to Lego blocks. In light of this, ASF is being repositioned as a suit of software tools. There will be eleven major elements in the suite:

Come and Get It

(1) ASF/OK Real Time Management of Objects
(2) ASF/GE Graphics and Events
(3) ASF/RM Removable Media Tools
(4) ASF/C Communications Tools
(5) ASF/XC Cross Compiler
(6) ASF/LC Standard C/C++ Libraries
(7) ASF/LL Common Lisp Object System
(8) ASF/XP Expert System Tools
(9) ASF/XU Expert Utilities
(10) ASF/DB Object Oriented Database Tools
(11) ASF/HL High Level language Macros

ASF/OK is the core of the suite and the other elements may or may not function without ASF/OK. ASF/OK will provide real time management of temporary and persistent objects, including doubly variable members. Each element will be written using C++ for reasons of portability. When running on non-x9000 hardware, file handling objects will need to substitute for the x9000.

ASF/OK will continue the theme of making life as easy as possible. The ultimate objective is to allow non-programmers to develop objects. ASF/OK will rely heavily on menus and lessons learned form ook. ASF/OK will support multiple inheritance and reverse inheritance, abstracting a generalized superclass from previously existing classes. ASF/OK will allow each class to specify member functions, called actors in this context, which tell the system how to implement system functions for the class.

I am planning to subcontract the development of the ASF elements. Since ASF Software is still an early stage effort I need to spread the risk. The obvious partial solution is to spread the revenue stream.

Attention Tool Builders

Calling tool builders. A future issue of Argonauts will be devoted to the available tools on the market for Amiga software developers.

They will be listed with a short description provided by you. The issue is intended to provide a convenient source of information to people just starting out in the Amiga market.

If you want your tool listed send a 50 to 75 word description and one working copy to Argonauts att: tools issue

Shareware Author's

Attention shareware authors. Argonauts is looking to put out a book on Amiga shareware. The book would be aimed at the typical consumer buying an A1200.

To be considered for inclusion the software must run on the AGA systems under AmigaDOS 2.1 or 3.x. Typical lead times for such a project are over 1 year so AGA based systems are the base line configuration.

We are specifically not looking for programmer utilities. Put yourself in the consumer's position. If your program doesn't seem useful than it wouldn't be included.

If you think you have something that would make the grade send one working copy to Argonauts att: shareware book.

No software will be bundled with the book. Only descriptions and author's contacts.

Comments and suggestions on the winding road to real time object management are encouraged.✔

ASF Software Corporation
303 North Rose Street Suite 315
Kalamazoo MI 49007-3847
800-832-4452

Late Breaking ASF News

Partial design has been completed on an ASIC implementation of the x9000 chip. This version includes an on board SCSI controller. Theoretically, you could use this part to design an Amiga expansion board with an integral hard drive. Running the ASF system directly on the x9000 would free up the Amiga's own cpu. This would turn the expansion board into a stand-alone object filing system. It would provide services to AmigaDos the same way that bridge boards do.

To promote interest and jumpstart the market for the ASF object oriented programming environment, the specification for the ASF Kernel has been placed in the public domain. It is available from ASF as an ASCII text file on MS-DOS diskette.

The ASF presents two opportunities for Amiga developers. The first is to use it to provide a generic persistent object filing system. The second and more interesting is to license the development system and port it to the Amiga.

The business model here would be to emulate what Digitalk did. They took an almost unknown language, Smalltalk, and used it to build a very successful company. They were able to do this on the technical strength of their product.Properly developed and integrated into the Amiga's GUI, the ASF system represents a similar potential for an Amiga developer.

A company like Innovatronics could use it to round out their product line. A more programer oriented product to complement CanDo. The compiler companies looking to broaden their suite of tools are also good candidates. A start-up looking to fill a currently empty niche would also be a possibility.✔

The Publisher

Vintage Software

Sometimes it pays to look outside your field for ideas, you never know what somebody else has come up with, especially if they have been around long enough. Vintners or wineries have been around for a very long time and they have learned a trick or two about selling. We're going to see if we can borrow some of their ideas.

Amiga Amiga Amiga Amiga
House Orion
Hyper-D

Chateau Amiga 92

Take a look at a wine label. What's the first thing you notice? The vintners logo. If it is an old line wine it will often be a chateau or castle. More recent vintners use the winery building or perhaps an abstract logo. What did you notice even before you looked at the label? You saw that it was a red, white or rose. From the shape of the bottle you knew that it was wine or champagne.

You absorbed all those things without conscious effort. The only thing that required detail work was spotting the name of the wine on the label to pick out your selection.

Compare this to typical software packaging. When I walk into a software store, there is nothing automatic about identifying software. Nothing helps me separate out the software into different categories, much less separate it by publisher or by machine type. Mac, PC and even Amiga software packaging is pretty much indistinguishable from each other.

Now in practice it is a little bit better. The retailer will usually separate out the software into machine specific sections. It still doesn't distinguish Amiga software from other machine types. Nor does it help find a particular product. Worse, this bookstore approach to displaying software does nothing to build brand loyalty.

About the Illustration

The Amiga fill pattern clues you this is Amiga software. The background grey color lets you know this is 3D modeling software. The oversize House Orion logo gives you the publisher. Finally, just beneath the manufactures name is the product name, Hyper-D.

Roll Credits

Think about what you see just before a movie starts. You see the logo of the company that made the movie. Go to the supermarket. Pick up a box of pasta. The brand name is likely to be the most prominent thing on the package. Look at cereal boxes. The name and logo of the parent company are very prominent. Look at a box of Arm & Hammer anything. The biggest symbol on the package is the company name.

A quick trip back to the winery and then we'll get to what all this has to do with computers. Wineries put the appellation "reserve" on the label of their very best wines. Sometimes there are only a few cases of this wine made. This is the stuff that wins prizes at competitions while the screw top bottles are what pay the bills.Vintners produce this wine for the same reason that Chrysler makes the Viper sports car. The aura around very high quality products builds a presumption in consumers that a little of what went into that is also in the more pedestrian product they buy.

The Amiga is finally selling at a million units a year, a number that ensures its long term survival. Even at this rate, it will always be the system with the smallest installed base. Consumers know the machine is different from the Mac or PC. The trick is to get them to perceive it as better, more advanced, available over a broader price spectrum, instead of as an also ran.

Collective Action

This is going to require some collective action. Collective action is often most easily achieved when you minimize what has to be done collectively. Here is the proposal.

Second Best

A common complaint leveled against the Amiga is that Microsoft, Lotus, Borland and WordPerfect products are not available for the system. This will remain true until two trend lines converge.

First, the installed base of AGA hard disk equipped machines will have to get into the 2 to 3 million range in the domestic U.S. market. Second, those companies will have to get so hungry for sales that they will scramble into this a market.

In the meantime, the Amiga can take a page from the Macintosh book. While having the native applications is everyone's first choice, data compatibility is a strong second. This data compatibility needs to be pursued on the foreign platforms (Mac, PC) as well as the Amiga.

Can the average Amiga wordprocessor open a MacWrite, Word or Wordperfect file? Can it save the file back out in those formats? The Macintosh floppy drive is the most versatile on any computer. Why can't I buy a Mac Init that would mount Amiga floppies the way Crossdos mounts MS-DOS floppies on an Amiga?

Adobe Postscript Type 1 and TrueType fonts are the dominant standards. Why aren't one or both natively supported on the Amiga? Why can't I buy Apple's Font Pack or Key Fonts and use them under AmigaDOS?

These are the kinds of questions non-technical Mac and Windows consumers ask about the Amiga. They usually don't phrase them that way but the core question is: "Can I use the data from the other machine?"

Instead of seeing this as a problem, Amiga vendors should look at this as an opportunity. The question of font support is up to Commodore, but the balance is in the hand of third party developers.

If you sell a wordprocessor provide for export and import of at least Microsoft Word and Wordperfect files. Once you have export/import working, go to a Mac company called DataViz and give them the translation codes for free. Why?

DataViz sells the most popular file translation package in the Mac market. It also works with PC's. The benefit of having your package listed as one of the file translations

You can put whatever you want on the back of your software package but on the front everyone follows some simple rules. Everyone hashes out a color scheme so that red is a game, blue is wordprocessing, gold is 3D ray tracing etc. The front of the box is decorated with a background of this color with the word Amiga repeated as a fill pattern. The only other markings will be the equivalent of the wine label.

A collective move to this type of cover art would make for a striking effect on the retail shelf. Amiga software would stand out immediately from other machine types. The color coded box cover would let them quickly zero in on the software category. The Amiga pattern in the background tells them it is Amiga software. Finally the distinctive labels would lead them to the brand they were looking for. A year after such a system was introduced people would wonder why it had ever been done differently.

Expensive Tastes

Finally we get to the "Reserve" labels. The Amiga's real advantages over competing systems not surprisingly show up on the high end systems. So why taylor some software just for the high end, something that will purposefully not run on the vast majority of the installed base?

To show people not just what the machine can do but what your company can do. By definition, software labeled reserve will always sell in small quantities. Its value is in the marketing cachet it gives you. Its value is in the leading edge development experience you get from creating the product. Remember that today's high end cpu is tomorrows entry level processor. Think of reserve software as a way of showing off.

One of the nice things about this whole idea is its small cost. As you exhaust existing stocks of packaging you phase in the new style. If you don't want to sell reserve software fine, the world needs jug wine just as much as the fancy high priced stuff. Implemented collectively this repackaging will have a very visual, very high impact effect. It is a start toward putting an end to the zero sum approach to marketing prevalent in the Amiga world.

It's something that helps you and your colleagues without hurting your own operation. That's as good a bargain as you will find.✔

Dealer's Corner

People comparison shop for computers just like they do for every other big ticket purchase. The apples to oranges comparison between different computer types is an opportunity for consumers to evaluate the Amiga. Your job is to provide them with the information they need to make an informed decision.

This information should be broad spectrum and as unbiased as you can make it. People don't like being hustled or pressured. Start with the system's history. It came out a year after the Macintosh. Illustrate the progression of machines right up to the 1200. You want to show them that they are buying into a continuum as opposed to a here today gone tomorrow system.

Talk about the various upgrades and trade in offers that Commodore has sponsored over the years. Contrast this to the Mac and PC vendor history of, "buy a new machine." The point here is not to start talking about ROMs and Agnus chips. The point is that the company has historically stood behind existing owners.

For the more technically inclined customers, you can point out the 4000 series model's replaceable cpu module. Unlike other systems, Mac or PC, the 4000 has been designed to allow the heart of the system to be changed. You can upgrade it not by adding an accelerator card but by replacing the main part of the system. Theoretically, you could replace the 68x cpu with a RISC based module using this hardware approach.

The advantage to the consumer is that they have a system that they will be able to keep current through several generations of product cycles. No other vendor offers this. Talk about the benefit of buying a system with integrated hardware and system software. Contrast this to the configuration and installation problems of making DOS, Windows and brand X hardware all work together. Point out how easy it is to add hardware. No dip switches to play with or arcane conflicts to work out among the boards.

Mention TV and movies that use or show the Amiga. Children who watch Nickelodeon or MST 3000 viewers will recognize the product. Bring them over to an Amiga display and show them some DCTV or AGA 256,000 color animations. Let them contrast the speed of operation between Mac or Windows PC and the Amiga for themselves.

Ask them if they own a camcorder. Show them what you can do if you have an Amiga. Give them accurate prices for low and high end software/hardware to do these things. Assume money is an object. Any machine can do anything when money is no object. Tell them to ask the local PC and Mac dealers for pricing to do comparable things.

Finally and only if they ask about such things get a little technical. The AGA machines with their 32 bit busses and special purpose chips will be subjectively two to three times faster than the Mac LC II or low end 386sx systems for typical operations. The AGA machines can display up to 256,000 colors in all resolutions vs. 32,000 for lo res and only 256 for hi res on the typical color Mac.

Don't be afraid to pitch the machine's strengths. If the consumer is looking for a home machine to finish up Lotus 1-2-3 or Wordperfect documents started at work the Amiga is a poor choice. If the they want high end desktop publishing it is also a poor choice.

What the machine is good at is integrating other consumer electronics that the customer already has. It goes great with their camcorder and VCR. The Amiga is for the creative mind. If you want to create your own 3-D landscapes, animations, videos, the Amiga can do it.

If you want to experiment with 3D creations you can never build in the real world, the Amiga can do it. Legos for adults and children with a

they handle would far out weigh any payments you might be able to get from them. By giving it to them you help raise the profile of your own products.

How about the AMAX people releasing a software package that would allow Amiga floppies to be mounted on a Mac? This seems like an excellent match of product idea and developer resources.

An idea for the Amiga Developers Association. Adobe Photoshop is pretty much alone in being able to directly import IFF image files. Why not collectively fund the development of code that Mac developers could embed in their file requesters? This code would allow IFF images to be opened/translated on the fly. Once written, this code could be made freely available to any Mac developer who wanted it.

Again why do this? Every Macintosh application that took advantage of this capability would be an advertisement for what can be done by Amiga applications (because that's where all those IFF images are coming from). Why would the Mac people bother to take the code? There's always someone looking to gain an advantage over a competitor Every extra feature they add gives them an edge, especially if they don't have to pay for it!

Any one of these things alone doesn't amount to much. Collectively, they would equalize the Amiga vis a vis the other platforms in data exchange.

Finally how about an ANIM to QuickTime converter? Animation and the software to produce it is clearly an Amiga strong point. Every QuickTime movie that descends from an ANIM file is a demonstration of what you can do with an Amiga. As CD-ROM evolves in the direction of user controlled "movies," the Amiga becomes a natural production platform.

If producers don't find the tools to easily get the data into the target machine's format, they will use other platforms. Each of these are business opportunities. It's up to the companies in the Amiga business to exploit them. ❑

beauty the originals can never match. This is an imagination machine that lets you build and create instead of watch and destroy.

Other machines can do these things too, but not as well nor as inexpensively. In short give them something to think about.✔

Newsline

ICOM Simulations Inc. has released "Sherlock Holmes, Consulting Detective" CD-ROM for the CDTV system. This product has sold over 250,000 copies in its various formats. Developers interested in joining the Adobe Developers Association can contact Adobe at 1- 800 833 - 6687. Membership is of particular interest to anyone using postscript in their products.

The Object Management Group has released The Common Object Request Broker: Architecture and Specification. A 177 pages, $50 contact 303 444 - 8129. Anyone writing Amiga network code should have a copy of this. Even if you have no interest in complying with the ORB standard it is useful for seeing how a large block of the computer industry has decided to do it.

Eastman Kodak has introduced a line of writable CD products at very competitive prices. Currently only PC's are supported with Macintosh support due this Spring. This product represents a business opportunity for an Amiga developer to build the software/hardware needed to use this device with the Amiga. For more information call Kodak at 1 800 - 242 - 2424 ext 52

On October 16 the New York Times ran a front page article titles, "Turning a computer Screen Into a Window on Whimsy." The article is a three column piece that makes the same point that the "Making it Mine" article in the Fall issue of Argonauts did. People want to put a personal imprimatur on their computers. The company profiled in the article, Berkeley Systems had sales of $15 million in 1991 mostly due to sales of the After Dark screen saver program.

3DO for three dimensional optics introduced its CD-ROM player at January CES. This company was founded by Trip Hawkins of Electronic Arts fame. 3DO is backed by Time Warner, Matsushita, MCA, AT&T and Kleiner Perkins. The system is based on the same RISC processor Apple uses for its Newton PDA. It will also include a high speed CD-ROM drive. It will play audio disks and display Kodak Photo CD's as well.Expected price is $800. The 3DO information excerpted from NYT articles on 12/19 and 1/8. □

Derivative Software

If you stay in business long enough, sooner or later you discover the secret to success. Make something once and sell as many variations as possible. Most entrepreneurs learn this trick quickly or they don't stay in business very long. Most people never think to carry it any further though.

Suppose you stepped back and thought about it. In physics when you take the first derivative you get a velocity function. Definitely something useful. But it doesn't stop there. You can take the second derivative and you also get something useful, the acceleration function.

Let's try and apply this to software. The Amiga has a plethora of 3D software aimed at the high end. The reality though is that Commodore will sell many more A600/1200 than A4000 computers. Is there an opportunity here?

Think about one of the most popular children's toys in the world, Legos. Now think about 3D modeling programs. If you are a little kid, don't you like to play what if games too? Wouldn't you like the fun of playing architect AND builder?

Go out and buy a reasonably complex Lego set. Look at the instructions. They are 3D cut away drawings that show you how to build the construction one layer at a time. Think about how many kids all around the world have Legos. Think about how many children in Europe where the bulk of Amiga's are sold have Legos.

Now think about a 3D product that allowed children to design their own constructions. A product that would allow them to look at it from different angles. A product that would then print out a series of cut away illustrations showing how to build the thing.

Can you spell add ons? Supplemental disk sets for an unlimited collection of structures. Supplemental disks that let you specify other than Lego blocks. Potential licensing deals and tie ins.

Different versions of the program. Say a basic one designed for young children up to 8 years old. A more advanced version that included the capability to calculate stresses on members. The child wants to build a cantilevered bridge or skyscraper. Let them use the program to experiment with just where the limits are and then test those limits by actually building the thing. This type program could easily become very popular in schools to say nothing of at home.

This program doesn't need super hi resolution, it doesn't need ray tracing and it doesn't need 16 million colors. Children's building blocks don't come in many more than 16 colors so there is no problem on that front. It needs to operate in the perspective view and produce accurate construction drawings. It needs to be fast, children don't like to wait. If you don't count rotating the structure, you don't need animation.

The manual should be a very short, over size(8.5 x 11) booklet less than 50 pages long with big type. Make it mostly illustrations and make it at least two color. When you design this product think TOY, CHILDREN, most of all FUN!

With all those 3D programs out there, how many Amiga vendors already have the core programming done for this product? So what are you waiting for. If you are really ambitious, get on the phone and see if you can get Lego to back you in creating this product.✓

Books

A quality product is much more than a bug free software program. It's a well thought out and executed user interface. It's an easy to install product. It's well written, readable documentation and tutorials. It's an attractive package. It's something with utility. It's something with aesthetics.

These books won't make you experts on these topics. To achieve that you need native talent and formal training. What they will do is make any project you work on better than it would have been without them. If you do this while your competition does not, it could be enough to give you that critical edge.

Technical Editing, The practical guide for editors and writers, Judith A. Tarutz

This is hardly a page turner but it is 400 plus pages of very solid advice on technical writing and editing. Whether you write the manuals or you edit them you will find this book useful. After reading it you may not write the great American novel but you will write and edit better manuals.

Looking Good in Print, Roger C. Parker.

Now that you've written an eloquent master piece you still need to format it into a manual. Looking Good is a heavily illustrated book that will make a difference even if you use it just to see what not to do. Anyone with enough skill to use a desktop publishing package can make visible improvements in the quality of their work by reading this volume. A definite must have.

TOG on Interface, Bruce "TOG" Tognazzini.

Run, do not walk to your nearest book store and buy this one. Tognazzini is such an interesting writer and the topic so fascinating that you will probably finish the 300 plus page book in one sitting. To all you programmers who think a user interface is a screen full of buttons READ THIS BOOK.

Envisioning Information, Edward Tufte

This book can be considered a sequel to Tufte's, "The Visual Display of Quantitative Information." Envisioning is best described as a pic-

About Confederacies

(1) To make confederacies work you need a neutral forum to present and promote your ideas.

(2) This forum must be inclusive. It cannot just be a domestic group.

(3) It must be depoliticised. By definition, a confederacy is the wrong place to resolve contentious issues.

(4) It should be in the forum's interest to remain unbiased.

Groups based on confederation that try to operate outside of these articles almost always fail. Confederacies, unlike federations, cede very little power. They are loose alliances based on self interest.

Argonauts was started because I saw a business opportunity in providing A forum that operates within these constraints. It's cheap. $50 a year so even the one man company can afford it. It's neutral, as a business the newsletter serves its purpose and pays its way or it eventually folds.

As a quarterly it arrives often enough to get information out in a timely manner but not often enough to be junk mail. The original premise was to focus on three areas: product development and research, international business linkups and challenge projects. Based on requests from developers, a fourth area will be added; economic self interest.

There are things that are in the self interest of many developers. They don't get done because there is no mechanism for getting them done.

The Amiga market is composed of many privately held firms. Depending on how everyone acts this can be a plus or minus. The market economy we all operate in is an organic entity, dependent on many different players:manufacturers, distributors, dealers etc. Argonauts is subtitled,"for the Amiga developer / entrepreneur," because we think it's in all our interests to form a chorus instead of a cacophony.

The "Vintage Software" article in this issue is a good example. Whether this particular format is adopted or not isn't important. Getting people to think about something that presents a more coherent marketing front toward the Mac and PC is. An inexpensive CD-ROM sampler with demo versions of all Amiga software minus games is something else in all our interests. Next issue we'll talk about low overhead mechanisms for getting these kinds of things done. ❑

The Publisher.

ture book. You don't so much read it as look at it. You pick up information from this book by osmosis. The images in this book don't show you how to solve your particular problem. What they do is build up patterns in your brain so that when you look at your own work alarm bells will go off. You keep making changes until the volume is lowered enough to be tolerable. A must have book.

The Nurnberg Funnel, John M. Carroll.

No where near as readable as TOG. This one you should borrow from the library. It approaches the topic of user interface from cognitive psychology. Carroll is a much less entertaining writer than Tognazzini but the material is worth reviewing even if you have to slog through it.

Designing and implementing a good user interface is much more than putting some 3D buttons on a screen. These two books will give you a feel for what interface design as about. It is very distinct and much more complicated than the task of laying out control features on the screen.✔

Research Corner

DSP chips are almost here on the Amiga. Hsiao-Wuen Hon's paper offer's a use for that power. Carnegie Mellon has arguably the best speaker independent voice recognition system around. Apple's Casper demonstrations are based on work done at CMU. Hon's paper is his research dissertation on the work that led to these systems. A definite must read if you are doing speech recognition software.

The other two papers deal with log based file (LFS) systems. This is a technique that creates what is probably the fastest file system you can implement when you are working with only one disk drive as opposed to disk arrays. The Berkeley paper describes an LFS for the SPRITE system. SPRITE is what Berkeley developed after the BSD project came to an end.

The Digital paper describes modifications made to that LFS. The DEC researchers describe an on-line compression system they added. This both increased the effective data transfer rate of the disk subsystem and increased its storage capacity. They also describe options for doing the compression in hardware instead of software.

With all new Amiga models now equipped with some kind of built in disk controller life is getting tougher for those selling controller cards. Combination software and hardware that replaced the Amiga's native FFS with an LFS combined with hardware compression and SCSI II controllers could lure buyers to spend on third party hardware.

A software only LFS product is also an interesting possibility. If it were a high quality implementation it could eventually follow AREXX and CROSSDOS into AmigaDOS.❑

Vocabulary-Independent Speech Recognition: The VOCIND System

Hsiao-Wuen Hon

CMU-CS-92-108

School of Computer Science

Carnegi^ Mellon University

Pittsburgh PA 15213

M. Rosenblum and J.K. Ousterhout The Design and Implementation of a Log-structured File System, Proceedings of the 13th ACM Symposium on Operating Systems Principles, Asilomar, 13-16 October 1991 pp 1-15.

See your local college library to get this paper or use inter-library loan from a public library if there is no nearby college

On-line Data Compression in a Log-structured File System

M. Burrows, C. Jerian, B. Lampson, T. Mann

Report # 85

Digital

Systems Research Center

130 Lytton Avenue

Palo Alto, CA 94301

Amiga Clock 4500000 sold

AL	3
AK	1
AZ	5
AR	2
CA	34
CO	5
CT	4
DE	2
FL	14
GA	3
HI	1
ID	0
IL	17
IN	10
IA	1
KS	4
KY	2
LA	5
ME	4
MD	5
MA	10
MI	11
MN	2
MS	1
MO	7
MT	1
NE	3
NV	1
NH	2
NJ	6
NM	3
NY	39
NC5	
ND	1
OH	15
OK	4
OR	8
PA	15
RI	2
SC	4
SD	0
TN	4
TX	14
UT	2
VT	
VA	8
WA	16
WV	1
WI	8
WY	0

US user groups by state, source CBM list

Commodore's 92 Annual Report

Net sales $911 million

Amiga accounted for $573.9 million

Amiga sales were 63% of total sales.

C64 accounted for $115 million

C64 sales were 13% of total sales

Balance of income was from PC sales misc. items.

Europe accounted for **88%** of total sales.

North America accounted for **8%** of sales.

Australia/Asia accounted for **4%** of sales.

Revenues by Region:

Europe 798.5 million

N. America 76.9 million

Australia/Asia 35.6 million

Yearly Amiga sales have reached the one million unit. Dividing by Amiga revenues, Commodore took in an average of $573.9 per Amiga sold.

While no unit sales figures are included in the report, N.A. sales were almost all due to Amigas. Dividing the average Amiga revenue by total N.A. revenues gets about 115,000 units.

While only C-A knows how close to the real number this is it is probably a ball park figure. Now you know where all the doom and gloom was coming from this past summer (CBM closes its fiscal year June 30).

The good news is that before the recession hit home, CBM had N.A. revenues of $163.5 million. With the revamped product line at both top and bottom look for 1993 to be much better than 92 was.

Useful Contacts

CATS Commodore Amiga Technical Support
1200 Wilson Drive
West Chester PA 19380

Association of Shareware Professionals
545 Grover Road
Muskegon MI 49442

US Copyright office
Request for forms 202 - 707 - 9100
Information 202 - 707 - 3000

AT&T 800 Consumer & Business Directory
To order 1 - 800 - 426 - 8686 ext 223

You always wanted to ask

The largest Amiga third party vendors are plus/minus in the $10 million range.

Many of the successful Amiga software companies are in the $2-5 million range. Successful is defined as running 4 color full page advertisements under your own banner.

You would be surprised at how many companies are in the $250,000 to 1 million range.

For the foreseeable future the Amiga market will be dominated by these size companies. Expect privately held firms to remain prominent for a long time to come.

The Competition Moves On

New Mac's to be introduced in Feb

LC III 4/80 25MHz 030 $1,379

Centris 610 4/80 20Mhz 040 $1,859

Centris 650 4/80 25MHz 040 $2,699

Source: Dec 28 PC Week and InfoWorld

Fine Print. This offer open only to subscribers. The publisher reserves the right to refuse any ad without explanation. This offer non-transferable. Limit of one submission per subscriber per year. Subscriber may submit a 60 word classified or a 2"x3" black and white camera ready ad. It will be run a minimum of once and a maximum of 4 times. There is no charge for this service. Submissions will be published on a first in first out basis. The intent of this offer is to aid companies in introducing new products and contacting each other. As in all business situations caveat emptor, let the buyer beware. The appearance of an ad does not constitute an endorsement by the publisher

Coming Soon

In the pipeline. First person business accounts by Matt Dillon and Merrill Callaway. A behind the scenes look at how ICOM Simulations put "Sherlock Holmes" CD-ROM together. Pie menus, research papers, new product ideas. Sarah Bell on Objective-C. News from Russia. Deep pocket markets for the Amiga. Sex and the CD

Argonauts
Box 94
Pearl River NY 10965 - 0094
USA